Rapport de stage technicien

ROUYRRE Rodolphe

Pour le 13/09/2019

Role : assistant dans la réalisation d'outils d'analyse prédictive

Période : du 17/06/2019 au 06/09/2019

Lieu : UTADEO, Carrera 4 #22-61, Bogota

Maitre de stage : Olmer Garcia Bedoya

fiche d'évaluation

Table des matières

1 Remerciements

Avant d'entrer dans le vif du sujet, je tiens à remercier toutes les personnes qui m'ont permis de faire ce stage, à commencer par le directeur de l'INSA de Rouen, M.Boukhalfa, l'ancienne directrice et l'actuel directeur du département Génie Mathématique, Mme.Zanni-Merk et M.Knippel ainsi que le dirigeant de l'université Jorge Tadeo Lozano, Jaime Pinzón López.

Je tiens à remercier particulièrement Edgar Mauricio Vargas, pour avoir répondu favorablement à ma demande, m'avoir conseillé idéalement pour les premières démarches administratives et de m'avoir accueilli dans le département qu'il dirige.

Je remercie également Ixent Galpin pour avoir pris la suite des démarches et validé ma demande de stage. De même, je remercie M.Kotowicz pour avoir levé mes interrogations et approuvé ce stage.

Bien évidemment, ce stage n'aurait été possible sans mon tuteur Olmer Garcia Bedoya. Je le remercie pour m'avoir encadré avec sérieux et bienveillance durant mon séjour.

Je remercie bien évidement les personnes qui ont travaillé avec moi : Edgar José Ruiz, Cesar Diaz Benito et Marvin Eliecer Viloria, professeurs et chercheurs sur le projet.

Je suis très heureux d'avoir pu réaliser mon stage d'exécution à l'université Jorge Tadeo Lozano aux cotés de tous ces employés et je garderai un très bon souvenir de cette expérience. Je remercie encore toute l'université pour le bon déroulement de ce stage.

2 Contexte

2.1 Global

Ce stage s'est déroulé du lundi 17 juin 2019 au 6 septembre 2019 dans l'université Jorge Tadeo Lozano. Plus précisément, durant celui-ci, je dépendais de la branche mathématique du département Ingénierie de la faculté Sciences Naturelles et Ingénierie.

2.2 Role

J'ai effectué mon stage technicien de fin de troisième année dans le département mathématique de l'université de Bogota Jorge Tadeo Lozano, courament appelée Utadeo ou Tadeo. Mon maitre de stage, docteur en ingénierie mécanique et professeur en automatisation, Olmer Garcia Bedoya, m'a encadré durant douze semaines.

Le travail qui m'a été assigné était de participer à un projet de 18 mois que l'université a mis en place en coopération avec l'UNAD, la Universidad Nacional Abierta a Distancia, littéralement « l'université nationale ouverte à distance ». Celui-ci a débuté en septembre 2018, je l'ai donc repris pour le onzième mois de travail.

Le but de ce projet est de créer des outils d'analyse statistique permettant la prédiction des cas de décrochage scolaire. En effet, l'UNAD étant une université en ligne, il est important pour les dirigeants de controler l'assiduité des étudiants.

Dans le cadre de mon stage, le but était d'apporter mon aide pour avancer sur le projet, c'est-à-dire analyser et établir la conception d'outils permettant une description statistique des données. J'ai donc fait mes premiers pas dans l'analyse de données, domaine à forte dominante informatique. Cette analyse a été réalisée avec les données de la Utadeo, qui utilise la même plateforme de LMS (learning management system), du nom de Moodle. Ces données, bien

qu'elles n'ont pas les mêmes caractéristiques sur les étudiants à distance, ont la même forme d'information et les types d'analyse statistiques et descriptions que l'on applique pour le projet.

Ce travail concerne donc le traitement d'informations obtenues à partir de la base de données de la plateforme informatique et l'analyse de celles-ci grâce à un langage informatique spécialisé dans la data science comme R ou Python. Etant plus familier avec le langage Python, j'ai choisi de travailler avec celui-ci. Ce stage avait donc une forte dominante informatique, impliquant les langages SQL et python, avec sa librairie Pandas, indispensable pour la data science. Ce projet fait également intervenir des notions basiques de statistique et de calcul scientifique. En effet, la plateforme en ligne générant plus d'un million d'accès par semaine, il est indispensable d'optimiser les opérations informatiques pour gagner en performance et en temps de calcul.

3 Présentation du cadre de travail

3.1 Présentation de la plateforme

L'université utilise la plateforme informatique moodle, sous le nom « AVATA ». Toutes les personnes de l'université y ont accès, aussi bien les professeurs que les étudiants. La création des cours se fait par les professeurs qui en sont ensuite les seuls administrateurs. Ils peuvent créer des activités que les étudiants devront valider. Les pages présentant les cours sont organisées en composants et en événements. Les étudiants sont donc obligés de se connecter pour valider leur cours. De plus, les professeurs, particulièrement ceux en informatique, encouragent les étudiants à se connecter à la plateforme durant le cours. Elle est donc très différente de celle que l'on connaît à l'INSA, les étudiants l'utilisent beaucoup plus, ce qui rend son analyse plus pertinente.

Le principe est le même pour les cours de l'UNAD, mais les étudiants sont également évalués sur la plateforme.

3.2 Présentation de l'université

3.2.1 Localisation géographique

L'université de Bogota Jorge Tadeo Lozano est située dans le centre de Bogota, capitale de la Colombie et du département de Cundinamarca.

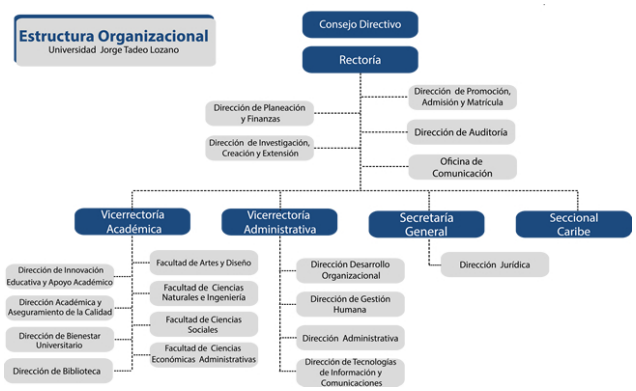
3.2.2 Présentation historique

Elle a été créée en 1954 par Joaquín Molano Campuzano, Javier Pulgar Vidal et Jaime Forero Valdés. Jorge Tadeo Lozano, né à Bogota en 1781 et mort dans la même ville en 1816, était un zoologue grenadien et a été le premier président de l'état libre de Cundinamarca. L'université a été nommée ainsi car

sa collaboration dans l'expédition botanique du Nouveau Royaume de Grenade a inspiré ses créateurs en raison des mérites universitaires et scientifiques qu'on lui doit. De plus, l'un des fondateurs, Joaquin Molano, en est le descendant.

Il existe également deux autres universités du même groupe (Utadeo) situées sur la côte Atlantique, au nord du pays, une à Cartagène et une à Santa Marta, fondées respectivement en 1976 et 1991.

Lors de sa création, l'université proposait 8 facultés, puis d'autres se sont ajoutées au fil du temps. Dans les années 2000, la structure de l'université a changé pour regrouper tous les cours au sein de 4 facultés nommées ainsi : arts et design, sciences économiques et administratives, sciences sociales, et sciences naturelles et ingénierie. Le département d'ingénierie dont je dépendais fait bien sûr partie de cette dernière.



Elle est composée de 9 batiments appelés modules. Aujourd'hui, elle dispose de 40 programmes universitaires, de licence, dont 16 agrégés de haute qualité par le ministère de l'éducation nationale colombien, et 53 programmes de doctorats. Au premier semestre de 2019, elle accueillait 8500 étudiants de premier cycle et 1500 étudiants diplomés en cours de spécialisation (doctorants, masters. . .).

3.2.3 Concurrence

On répertorie 41 universités à Bogota. L'Utadeo est classée sixième université privée de Bogota et quatorzième au niveau national selon QS World University Rankings. Les trois premières places pour Bogota sont occupées par l'université nationale de Colombie, l'université externe de Colombie et l'université Javeriana. Celles-ci sont trois des 5 autres universités situées au centre de Bogota, à moins de 10 kilomètres de la Tadeo. Cette zone est donc particulièrement concurrentielle en matière d'études supérieures. 360 000 étudiants sont répertoriés pour une population de près de 7,7 millions d'habitants.

3.2.4 Organisation

Le directeur du département d'ingénierie est le professeur Edgar Mauricio Vargas, que j'ai pu côtoyer. J'ai également travaillé avec Olmer Garcia Bedoya, mon maitre de stage et professeur en ingénierie des systèmes et spécialisation en développement de base de données. Les autres personnes en lien avec ce projet étant Ixent Galpin, Marvin Eliecer Viloria et Cesar Diaz Benito. Enfin, Sergio Morales Dussan, étudiant en master en modélisation et simulation et chercheur pour le projet mentionné, a pris en charge la suite de mon travail.

3.3 Présentation de l'UNAD

Ce projet est le fruit de la collaboration de mon université d'accueil, l'Utadeo, et l'UNAD, une université proposant des cours à distance et utilisant la même plateforme que la Tadeo. C'est une université en ligne créée en 1982. Le projet auquel j'étais relié était réalisé uniquement pour le département géographique d'Antoquia de l'université.

4 Développement du travail traité

4.1 Présentation du travail demandé

J'ai analysé les accès des étudiants de l'université à la plateforme informatique de celle-ci.

Tout d'abord, mon maitre de stage m'a présenté le projet, ce qui avait déjà été fait et les tâches qui m'étaient assignées. En effet, le projet ayant commencé en septembre 2018 (voir chronogramme en annexe), il était déjà bien avancé et l'UNAD avait présenté ses premières observations.

L'UNAD avait donc déjà établi un rapport d'une vingtaine de pages, présentant des résultats comme les variables ci-dessous :

Variable	Description
Tp	Total time spent by student in the CMS
Tt	Average time spent by a student in an activity
Ef	Average posts in forums for each student
Dp	Average time (days) spend by each student to participate in an activity
De	Average time (days) spend by each student to send the activity products
Vm	Number of visits to the course materials
Nl	Number of logs generated by the student
Ls	Number of weekday generated logs
Lf	Number of weekend generated logs
Ld	Number of diurnal generated logs
Ln	Number of nocturnal generated logs
Pa	IP ratio
Fa	Access frequency

TABLE 4. Logs characterization for each student

J'ai également été convié à plusieurs réunions afin de savoir quelles étaient les problématiques du projet et les difficultés auxquelles devaient faire face les professeurs.

4.2 Outils utilisés

4.2.1 Généralités

L'université m'a fourni un poste de travail dans un bureau en accueillant 6, les 5 autres étant occupés par des professeurs. Il était à proximité du bureau du professeur Garcia Bedoya et le poste était équipé d'un ordinateur sous Ubuntu.

4.2.2 Python

Après lui avoir rappelé mes compétences, mon maitre de stage m'a suggéré d'utiliser le langage Python, fortement utilisé pour l'analyse de données grâce à la librairie Pandas.

La librairie Pandas : Cette librairie est particulièrement utile pour créer des dataframes, c'est-à-dire des tableaux indexés où toutes les colonnes portent un nom. Celles-ci ont l'avantage de pouvoir être créées à partir de la table d'une base de données mais aussi d'un fichier avec l'extension csv. Pandas dispose également de nombreuses opérations sur ces dataframes particulièrement puissantes. N'étant pas familier avec cette librairie, il était intéressant de commencer par la découvrir à partir de sa documentation et de divers tutoriels.

Le module Pylatex : J'ai utilisé le module Pylatex pour pouvoir écrire des rapports en LaTeX à partir de Python dans le but d'automatiser leur rédaction.

Le logiciel Spyder : Mon tuteur m'a également conseillé de travailler avec le logiciel Spyder, car il facilite grandement la compilation de code. Il dispose d'une console permettant l'exécution de quelques lignes ce qui est particulièrement utile pour ne pas avoir à exécuter le code à chaque fois. Cela me permettait également de vérifier la syntaxe de Python et des opérations de Pandas dont je n'étais pas certain.

En bref, j'ai aussi utilisé les modules suivants :
— Numpy : pour créer des tableaux
— Matplotlib : pour créer des graphiques
— Os : pour la gestion de fichiers et de répertoires (déplacer, créer, renommer)
— Seaborn : pour créer des cartes de chaleur
— Time : pour utiliser le temps relatif à l'ordinateur
— Datetime : pour gérer les données temporelles

4.2.3 SQL

J'ai dû apprendre les bases du langage SQL. En effet, il n'était pas nécessaire que j'apprenne les commandes utiles pour la création ou la modification d'une base de données, mais celles pour la sélection et recherche de données m'étaient indispensables. J'ai donc été amené à me plonger dans la documentation pour savoir comment me connecter à la base de données à partir de Python et recueillir les données qui m'étaient nécessaires. Les commandes qui m'étaient utiles étaient donc SELECT, WHERE, JOIN et ORDER BY. Cependant, pour

mieux comprendre la base de données, je me suis également renseigné sur les notions d'index et de clés primaires et étrangères.

Le système de gestion de base de données que j'utilisais était MySQL. Je m'y connectais à partir de Python grâce au module MySQLdb.

4.3 Déroulement du travail

4.3.1 Création du premier programme

Avant de travailler sur la base de données, j'ai commencé par analyser les données pour le cours de mon maitre de stage. Celles-ci étaient présentées dans un fichier avec l'extension csv contenant les mêmes informations que la base de données. Cela m'a permis de développer mes connaissances en Python sans avoir à me soucier de recueillir les données grâce au langage SQL. Le fichier personnel de mon maitre de stage différait cependant légèrement de ce que l'on peut obtenir à partir de la base de données. Effectivement, les noms des colonnes étaient différents, les noms des composants et la date de connexion également. Les colonnes étaient les suivantes (j'ai laissé les noms en espagnol pour rester fidèle et je pense qu'ils sont assez transparents) :

- Hora : contenant la date et l'heure de connexion au format année/mois/jour heures :minutes
- Nombre del usario : contenant le nom de l'utilisateur avec les accents
- Usario afectado : contenant, s'il existe, l'utilisateur affecté
- Componente : contenant les noms des composants
- Nombre Evento : contenant les noms des événements
- Descripción : contenant une brève description de l'activité
- Origen : contenant l'origine de la connexion (web, restore, cli)
- Dirreción IP : contenant les adresses IP

Grâce à la librairie pandas, on peut créer un dataframe à partir d'un fichier csv. Il est ensuite aisé de travailler sur ce dataframe en indiquant le nom des colonnes ou l'indice de la ligne. Cependant, pour la cohérence du projet, il est nécessaire que les colonnes utilisées aient le même nom dans tous les programmes. En effet, il est pratique de diviser le code en fonctions contenues dans des modules. Celles-ci travaillent sur les colonnes des dataframes avec des noms bien précis que le dataframe doit contenir lorsque ces fonctions sont appelées. Il m'était donc impératif de choisir un nom clair et fixe pour les colonnes de mon dataframe. De plus, le langage Python ne comprend pas les accents et les caractères spéciaux, qui provoquent également des erreurs lors de l'affichage. Or, l'université étant située en Colombie, beaucoup de noms d'étudiant en possèdent, il est donc nécessaire de les retirer.

Finalement, le dataframe sur lequel j'ai travaillé contenait les colonnes suivantes :

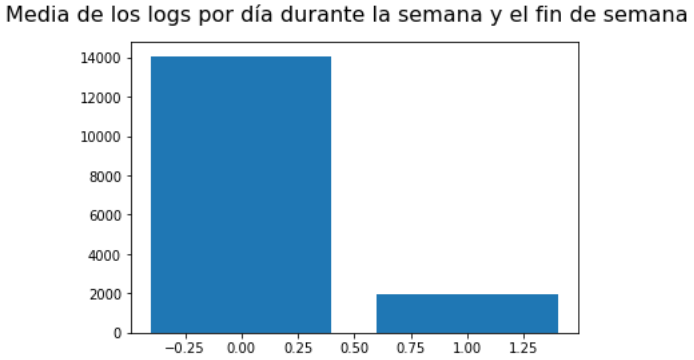
- User full name : contenant les noms des utilisateurs sans accent
- Component : contenant le nom des composants sans accent

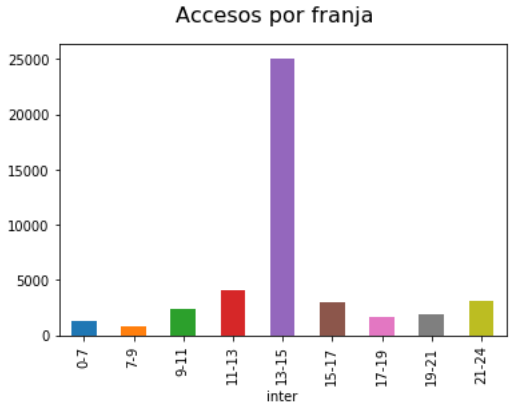
— Event context : contenant le nom des événements sans accent
— Fecha : contenant la date au format année-mois-jour
— Hora : contenant l'heure de connexion au format heures :minutes
— Dayofweek : contenant des chiffres de 0 à 6 indiquant le jour de la semaine (0 pour lundi, 6 pour dimanche)
— DiurNoct : contenant 0 ou 1 indiquant si la connexion a été établie le jour (1, avant 18h) ou la nuit (0, après 18h)
— Franjas : contenant des chiffres de 1 à 9 représentant les plages horaires

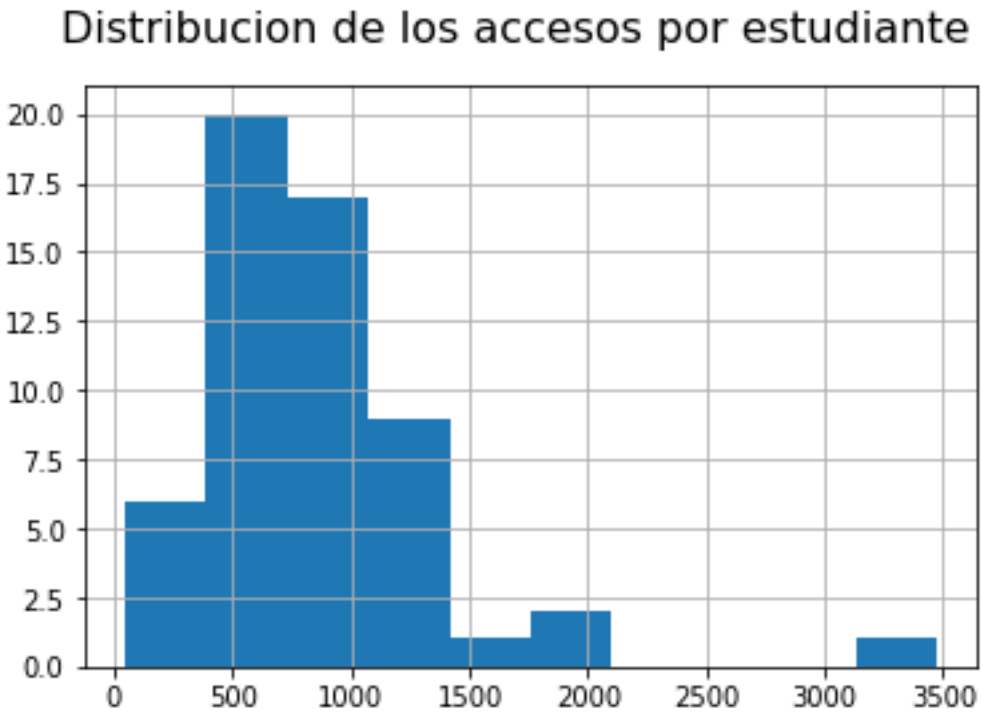
J'ai choisi ces noms car ils sont proches de ceux des colonnes de la base de données (voir diagramme).

4.3.2 Implémentation des histogrammes

Les premières semaines de mon stage ont donc consisté en l'élaboration de fonctions permettant de réaliser des histogrammes et cartes de chaleur pour visualiser les données. J'ai donc été amené à analyser les principaux composants et événements, le nombre d'accès par étudiants, par jour de la semaine, par intervalle de temps et la distribution du nombre d'accès par étudiant. J'ai également comparer les accès selon qu'ils aient lieu durant la semaine ou la fin de semaine ainsi que s'ils aient lieu durant la nuit ou la journée. Voici quelques exemples d'histogramme que j'ai pu réaliser :







Il y a plusieurs manières de tracer des histogrammes en python. La plus simple est d'utiliser la fonction bar du module matplotlib. C'est la méthode que j'ai utilisé pour coder les histogrammes pour lesquels les valeurs des batons étaient calculées au sein du code et non déjà présentes dans un dataframe.

```
# Medias por dia
MS = totS/(5-len(diasDeClase[m]))
MW = totW/2

t = [MS,MW]

#Bar graphico de las medias de logs durante la semana y el weekend
fig, ax = plt.subplots()
plt.bar(range(2),t)
fig.suptitle('Media de los logs por día durante la semana y el fin de semana', fontsize=16)
plt.subplots_adjust(top=0.88)
fig.savefig('Hist_Semana-Weekend{0}.png' .format(m2), dpi=fig.dpi, bbox_inches='tight')
```

Une autre manière de faire est d'utiliser la fonction pré-implémentée de Pandas : plot. Le résultat est similaire mais cette méthode est beaucoup plus simple lorsque l'on travaille avec des dataframes, c'est pourquoi c'est celle que j'ai choisi d'utiliser pour créer la plupart des histogrammes.

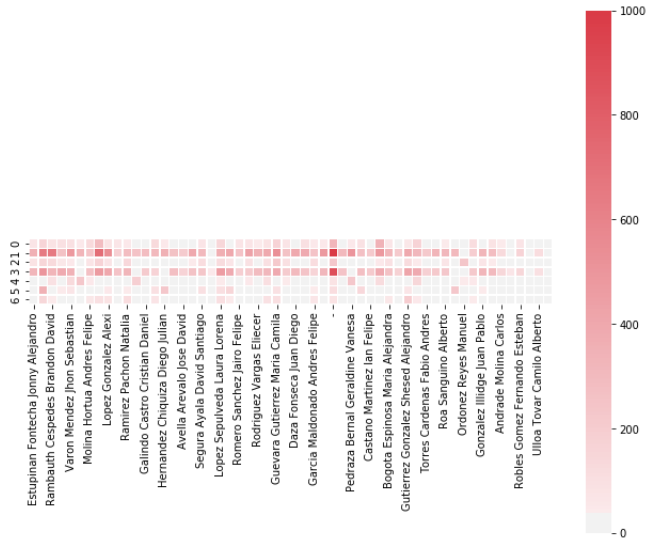
```
@controlar_tiempo()
def histFranjas(logs, m=0):
    """La función hace un histograma de los numeros de acceso en función de franjas de tiempo.
        """

    inter = ['0-7','7-9','9-11','11-13','13-15','15-17','17-19','19-21','21-24']

    dfFranja = pd.DataFrame({'Value': logs.groupby('Franja').size(), 'inter': inter})

    dfFranja = dfFranja.set_index('inter')

    # Histograma de accesos por franja
    fig, ax = plt.subplots()
    dfFranja['Value'].plot(kind='bar')
    fig.suptitle('Accesos por franja', fontsize=16)
    plt.subplots_adjust(top=0.88)
    fig.savefig('Hist_Franjas{0}.png' .format(m), dpi=fig.dpi, bbox_inches='tight')
```

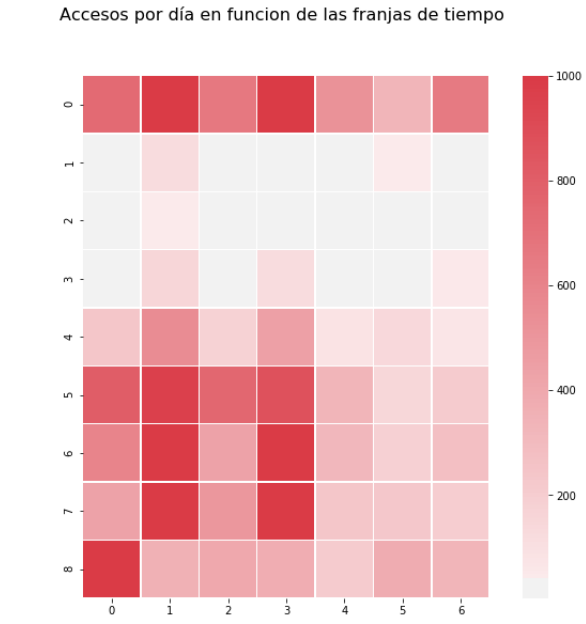
Pandas dispose également d'une fonction hist, directement implémentée sur les dataframes. Celle-ci est pratique lorsque l'on travaille sur des dataframes mais je trouve le résultat moins esthétique. Seules les distributions ont été réalisées ainsi car c'était la méthode la plus simple et la plus efficace pour les tracer. Elle permet de réaliser celles-ci de manière vectorielle, avec un faible nombre d'opérations et donc un moindre coût de calcul. Sans elle, j'étais contraint d'utiliser une boucle for sur les 50 étudiants du cours, ce qui est déconseillé étant donné que le but final est de faire tourner le programme pour analyser les 100 cours de l'université.

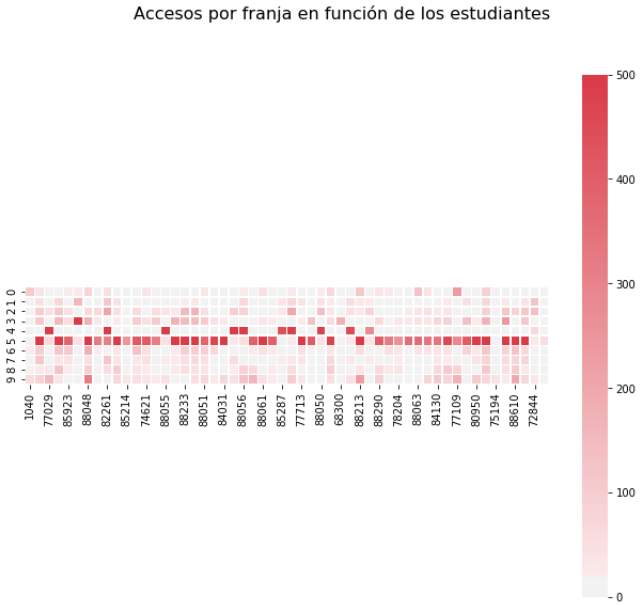
```python
@controlar_tiempo()
def histDistStud(logs, m=0):
    """La función hace una histograma de la distribución de los accesos por estudiante.
        """

    dfStud = pd.DataFrame({'Value': logs.groupby('User full name').size()})

    # Distribución de los numeros de accesos por estudiante
    fig, ax = plt.subplots()
    fig.suptitle('Distribucion de los accesos por estudiante', fontsize=16)
    dfStud['Value'].hist(bins=10)
    plt.subplots_adjust(top=0.88)
    fig.savefig('Hist_Distribucion{0}.png' .format(m), dpi=fig.dpi, bbox_inches='tight')
```

4.3.3 Implémentation des cartes de chaleur

Les cartes de chaleur ne sont réalisables que pour les colonnes présentant des valeurs numériques. Cela était donc possible uniquement avec les colonnes Dayofweek et Franjas. Avec celles-ci, j'ai donc pu réaliser les cartes de chaleur suivantes, représentant respectivement le nombre d'accès par jour en fonction des étudiants, le nombre d'accès par intervalle de temps en fonction du jour de la semaine et le nombre d'accès par intervalle de temps en fonction des étudiants :





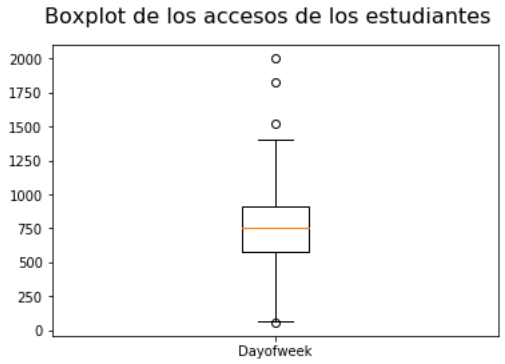


Pour coder ces cartes de chaleur, je me suis beaucoup inspiré de résultats trouvés sur internet. Le principe est de créer un dataframe aux dimensions désirées et indexé correctement mais rempli de manière aléatoire, puis de remplir les colonnes une par une grâce à la fonction histogram de numpy.

La carte de chaleur est ensuite tracée à l'aide de la fonction heatmap de seaborn, prenant comme paramètre le dataframe et l'échelle pour la variation des couleurs.

```
@controler_temps()
def heatmapDayofweekStud(logs, m=0):
    """La función hace un mapa de calor de los numeros de acceso por estudiante en función del día de la semana.
        """

    dfUser = pd.DataFrame(data=rs.normal(size=(7, len(logs['User full name'].unique().tolist()))),
                          columns=logs['User full name'].unique().tolist())

    # Para sacar los datos
    for name in logs['User full name'].unique().tolist():
        l = (logs['User full name']==name)
        l2 = logs[l]
        dfUser[name] = np.histogram(l2['Dayofweek'],bins=7)[0]

    # Mapa de calor de los accesos en función del día y de los estudiantes
    plt.figure()
    fig, ax = plt.subplots(figsize=(11, 9))
    fig.suptitle('Accesos por día en función de los estudiantes', fontsize=16) # Para poner un título
    cmap = sns.diverging_palette(220, 10, as_cmap=True)
    sns.heatmap(dfUser, cmap=cmap, vmax=1000, center=0,
                square=True, linewidths=.5)
    plt.subplots_adjust(top=0.88)
    fig.savefig('Heatmap_Dayofweek{0}.png' .format(m), dpi=fig.dpi, bbox_inches='tight')    # Para guardar el mapa de calor
```

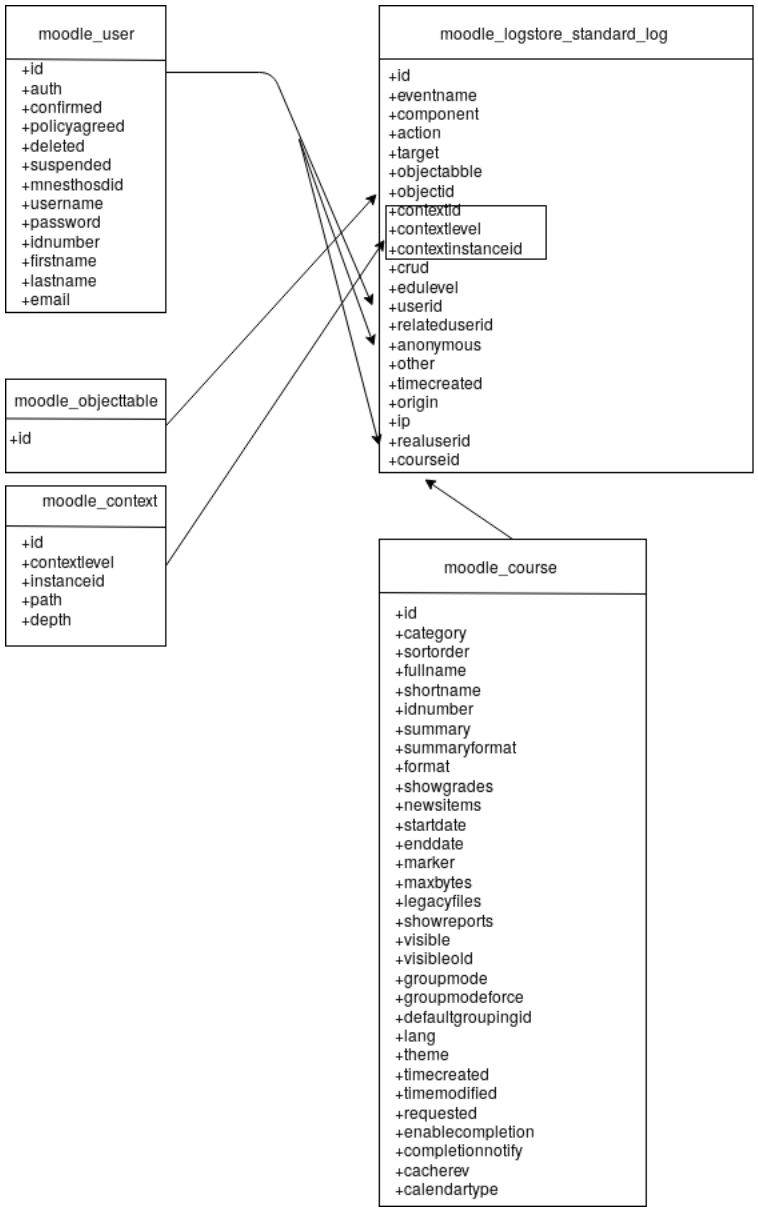
J'ai également automatisé la création d'une boite à moustaches pour pouvoir visualiser rapidement les caractéristiques des accès des étudiants au cours sur la plateforme.



4.3.4 Création du diagramme de la base de données

Ensuite, il fallait que j'utilise la base de données de l'université. J'ai donc dû réaliser un diagramme de celle-ci pour me familiariser avec elle et pouvoir y rechercher les données plus facilement. Cela est une étape indispensable pour connaitre les principales tables, les colonnes de celles-ci ainsi leurs liens. En effet, il est ensuite plus facile de rechercher des données grâce à des jointures en

se référant à ce diagramme.



4.3.5 Création du deuxième programme

La suite logique de mon travail était donc de reprendre mon programme et de l'adapter de sorte à ce qu'il génère les mêmes résultats, mais cette fois-ci à partir de la base de données. Mon maitre de stage m'a donc fourni un nom

d'utilisateur et un mot de passe me permettant de me connecter à la base de données. Grâce à ceux-ci j'étais en mesure de m'y connecter depuis Python et d'y rechercher des données pour les extraire, mais je ne disposais bien sûr pas des droits nécessaires pour modifier la structure de la base ni même ses données.

On peut créer une connexion « conn » grâce à la ligne de code suivante :

conn = MySQLdb.connect('nom de la base', user= 'nom de l'utilisateur', password='mot de passe')

Depuis celle-ci, on peut créer un « cursor » avec la commande :

cursor = conn.cursor()

C'est à partir de celui-ci que l'on peut utiliser des fonctions intéressantes pour sortir les données comme la fonction fetchall.

Enfin, la fonction de Pandas permettant de créer un dataframe à partir depuis une table se nomme read_sql_query et prend en paramètres la commande en langage SQL et la connexion « conn ».

Une modification pour adapter le programme était de créer les colonnes nécessaires qui n'étaient pas présentes dans la base de données. En effet, dans celle-ci la seule donnée temporelle est la colonne timecreated qui contient la date et l'heure de connexion à la base de données sous le format timestamp. Celui-ci est un format temporel pour lequel un nombre représente le nombre de secondes s'étant écoulées depuis le 1er janvier 1970 à 00h00. J'ai donc dû créer toutes les colonnes temporelles en précisant le format désiré. J'ai séparé la date en deux colonnes : une contenant le jour et l'autre l'heure au format datetime. A partir de celle-ci, il était désormais possible d'utiliser les fonctions du module datetime pour créer les colonnes 'Dayofweek' et 'Franjas' par exemple.

Une difficulté a été de sortir les professeurs du dataframe pour ne pas les compter et ainsi analyser uniquement les connexions des étudiants. Je n'avais pas de liste des professeurs et il fallait de toute façon automatiser cela pour chaque cours. Une façon de faire ceci a été de regarder les utilisateurs qui en affectaient d'autres. En effet, la colonne relateduserid, où se trouvent les utilisateurs affectés, contient NULL la plupart du temps, mais lorsque ce n'est pas le cas, c'est qu'un professeur s'est connecté en affectant quelqu'un d'autre.

4.3.6 Implémentation de l'automatisation des rapports

Une tâche qui m'était demandée était d'automatiser des rapports pour les cours que j'analysais. Mon maitre de stage, m'a incité à créer une fonction permettant de regrouper les figures (histogrammes et cartes de chaleur), dans un même document LaTeX, accompagnées chacune d'un commentaire automatique. J'ai donc dû regarder la documentation du module Pylatex pour réaliser de bons rapports en latex à partir de Python. En plus de cela, il me fallait sauvegarder les figures dans un dossier précis et m'y référer en cas de besoin. Cela était possible grâce aux fonctions du module os.

4.3.7 Création du troisième programme

Ensuite, il était intéressant de réaliser ce même programme mais en demandant à l'utilisateur d'entrer les informations nécessaires. En effet, cela permet à un professeur de générer une analyse personnalisée de son cours simplement en entrant le nom de celui-ci, le(s) jour(s) de cours et le nom des professeurs. Ce programme permet aussi de traiter intelligemment les exceptions, en prenant en compte tous les cas possibles grâce au mot-clé except. Il permet de supprimer les professeurs des données en se référant directement à la base de données, ce qui est beaucoup plus rapide. Le fait de renseigner les jours de cours permet de générer un meilleur rapport, plus personnalisé, sans ajouter de coût de calcul superflu.

4.3.8 Création du quatrième programme

La dernière partie de mon stage consistait à reprendre les fonctions et programmes que j'avais créé et les modifier pour analyser cette fois-ci, chaque semaine de la base de données. En effet, mon maitre de stage trouvait cela intéressant d'avoir une analyse globale de la base de données avec tous les cours. Cela permet également d'analyser l'activité sur la plateforme d'un point de vue temporel et de voir son évolution au cours d'un semestre ou d'une année. La plateforme générant énormément de connexions, je ne pouvais traiter qu'une semaine à la fois. Effectivement, même pour une période aussi courte, il me fallait restreindre le nombre de lignes à 500 000 pour être sûr que le programme ne prenne pas trop de temps. J'ai donc repris les histogrammes et cartes de chaleur que j'avais codés, exceptés ceux prenant en compte la colonne 'User full name' car il n'était pas intéressant de visualiser les 500 000 étudiants. Il était par contre utile de visualiser les principaux cours et la distribution du nombre d'accès par cours.

Bien que je n'ai traité que quelques cours et qu'une semaine à la fois, le projet a prévu la création d'un serveur dédié aux analyses de ce genre pour l'université. Avec celui-ci, les 100 000 cours pourront être analysés ainsi que n'importe quelle période de temps.

4.3.9 Partage de mes programmes

Ne pouvant assurer de travailler que pour 3 mois sur le projet, il était nécessaire que quelqu'un reprenne mes programmes pour qu'ils n'aient pas été réalisés en vain. A la fin de mon stage, j'ai donc fait part de mes avancées à Sergio Morales Maussan, chercheur sur ce-dit projet. Sachant cela, il était important que mon code soit :

- Organisé : divisé en fonctions placées dans des modules, se trouvant dans un package
- Lisible : indenté et aéré correctement

— Compréhensible : commenté de manière complète et explicite et équipé de docstrings pour comprendre rapidement les fonctions

Enfin, j'ai également écrit un commentaire sur mes programmes pour que n'importe qui puisse les reprendre sans avoir à me contacter. Celui-ci contient donc une notice d'utilisation des programmes et leurs améliorations possibles.

4.3.10 Optimisation du code et difficultés rencontrées

Une difficulté importante pendant mon stage a été de réduire au maximum le temps de calcul. En effet, je me suis moi-même rendu compte que je perdais beaucoup de temps à exécuter les programmes lorsqu'ils n'étaient pas optimisés. Au début de mon stage, je préférais utiliser des boucles for parcourant toutes les lignes de mes dataframes pour analyser les données, car cela était plus simple et je ne connaissais pas d'alternatives. Or, mon maitre de stage m'a rapidement corrigé en me conseillant d'utiliser les fonctions de Pandas.

Effectivement, l'intérêt de cette librairie est d'effectuer toutes les opérations vectoriellement, c'est-à-dire sans utiliser de boucle, pour gagner en performance et en temps de calcul. Le premier fichier contenait 50 000 lignes, mais par la suite j'ai dû travailler sur des dataframes de 500 000 lignes, pour des raisons de temps de calcul évidentes, il est donc très fortement déconseillé de parcourir ces lignes avec une boucle. J'ai été amené à utiliser un décorateur controlant le temps pour m'assurer que mes fonctions n'en prennent pas trop.

Une dernière difficulté a été de changer d'afficher les noms des variables sur les diagrammes. Pour des colonnes ne contenant pas beaucoup de valeurs (comme les colonnes 'Franjas' ou 'Dayofweek'), il était facile de créer une liste les contenant et de prendre celle-ci en tant qu'index du dataframe créé pour réaliser l'histogramme. En revanche, cela était impossible pour mettre le nom des étudiants. En effet, dans la table principale, ils n'apparaissent que sous forme d'ID. La seule manière de me référer à leur nom était de les sélectionner à partir de leur ID dans une autre table, celle des utilisateurs. Dans celle-ci je pouvais sélectionner leur nom et prénom, puis les concaténer et bien sûr les stocker dans une colonne du dataframe.

Le problème était le même pour afficher les principaux cours. Il est important de savoir que rechercher ainsi des cases précises dans la base de données allonge considérablement le temps d'exécution de la fonction, même pour une boucle de 10 itérations seulement.

En outre, j'ai bien sûr dû m'habituer à parler et écrire en espagnol, mais aussi à travailler avec un clavier colombien.

5 Résultats

Au cours des 12 semaines, je présentais mes résultats à mon maitre de stage régulièrement, dès que les tâches qu'ils m'avaient données étaient réalisées. Il me donnait les prochaines et ainsi de suite. Je tentais bien sûr de faire preuve d'un maximum d'autonomie, mais du fait de ma relativement faible connaissance du projet et des langages (du moins au début), ces réunions étaient indispensables.

5.1 Modules

La production de ces 3 mois de travail, est la création de 4 programmes et autant de modules. Ces derniers sont nommés hist, heatmap, informe et otros. Le premier contient l'ensemble des fonctions, qui sont au nombre de 10, permettant de créer des histogrammes. Dans la même logique, le deuxième contient les fonctions réalisant des cartes de chaleur et le troisième celles chargées de générer les rapports. Le dernier module, otros, contient le décorateur controlant le temps d'exécution d'une fonction, la fonction créant la boite à moustaches et une autre calculant des variables utiles pour d'autres fonctions, comme le nombre d'accès par étudiants.

5.2 Programmes

J'ai réalisé 4 fichiers Python pour analyser les données. Un pour analyser un cours à partir d'un fichier avec l'extension csv contenant les informations du cours, un autre réalisant la même chose à partir de la base de données, puis ce même programme mais cette fois en donnant la possibilité à l'utilisateur d'entrer lui-même les informations nécessaires pour analyser le cours qu'il souhaite. Le dernier fichier permet d'analyser une semaine d'utilisation de la base de données, avec tous les cours, pour avoir une idée générale de celle-ci.

5.3 Interprétations

L'exécution de ces programmes permet déjà de proposer certains résultats intéressants. Tout d'abord, les professeurs peuvent avoir une idée générale des connexions réalisées sur la page de leur cours au sein de la plateforme, comme les principaux événements et les principaux composants. Plus globalement, j'ai pu remarquer que, pour la plupart des cours, la majorité des connexions générées étaient réalisées durant les jours où le cours a lieu. Une telle analyse descriptive permet donc de mettre en évidence ce biais qui est à prendre en compte dans les futures analyses. Enfin, n'ayant pas accès aux notes des étudiants je n'ai pas pu analyser une éventuelle corrélation entre leur comportement vis-à-vis de la plateforme informatique et leurs résultats scolaires, mais un de leurs professeurs peut se faire sa propre opinion.

5.4 Apport pour l'université

L'analyse statistique descriptive que j'ai effectuée apport un nouveau regard au projet. Elle permet une visualisation générale des données pour avoir une idée globale de celle-ci. Cela permet de connaitre les points potentiellement intéressants à analyser. De plus, cela permet de relever des biais à éviter comme celui dont j'ai parlé au-dessus.

De plus, mon travail est déjà utile pour les professeurs souhaitant analyser leur cours.

6 Conclusion

6.1 Bilan

Les programmes que j'ai réalisé permettent une première analyse statistique descriptive comme cela était demandé. Le but initial étant simplement d'avancer sur un projet en cours, je n'ai pas eu de cahier des charges précis au début de mon stage. Il est donc difficile d'évaluer la qualité du rendu final de celui-ci. Cependant, le retour de mon maitre du stage sur mon travail a été très positif et celui-ci a été à la hauteur de ses attentes.

Néanmoins, je suis conscient que les résultats de mon travail restent assez basiques et je regrette de ne pas avoir pu rester plus longtemps pour tester mon programme sur le serveur de l'université. De ce fait, j'essaierai de suivre l'avancée du projet autant que possible.

6.2 Améliorations possibles et futur du projet

Durant ces douze semaines, j'ai tenté de réaliser mon travail du mieux possible. Cependant, il reste bien évidemment des améliorations possibles aux programmes que j'ai créés. Tout d'abord, je n'ai pas réussi à convertir le format des dates de manière vectorielle. En effet, la fonction to_datetime de Pandas ne donnait pas les résultats attendus. De plus, les cartes de chaleur ne sont pas toutes légendées de sortes à ce que les noms explicites des variables apparaissent et non leur ID.

6.3 Apport personnel

Personnellement, ce stage m'a permis d'atteindre toutes les compétences que je souhaitais développer lorsque l'on m'a présenté le projet, c'est-à-dire apprendre le langage Python avec ses librairies Pandas et Pylatex, la sélection et recherche de données dans une base de données grâce au langage SQL et le langage Latex pour la rédaction de rapports scientifiques. En effet, le niveau acquis dans ces langages dépasse mes espérances initiales.

D'une manière plus générale, je pense avoir pu développer mes compétences en informatique et ma manière de coder. Effectivement, travailler ainsi dans une équipe scientifique m'a obligé à suivre des règles de codage pour le bon déroulement d'un projet.

Le sentiment qui m'accompagne à la fin de ce stage est donc extrêmement positif, c'est pourquoi ce dernier m'a conforté dans mon intention de travailler dans l'analyse de données.

7 Grille de déroulement du stage

Semaine à partir du :	Tâche
17/06	Introduction au projet et première approche à la librairie Pandas
24/06	Création des premiers histogrammes à partir du fichier csv
01/07	Création des premières cartes de chaleur et des autres histogrammes
08/07	Introduction au langage SQL. Réalisation du diagramme de la base de données
15/07	Création du nouveau programme se connectant à la base de données
22/07	Implémentation de l'automatisation des rapports
29/07	Implémentation des figures relatives aux intervalles de temps
05/08	Adaptation du code pour créer le programme pour les professeurs
12/08	Création du programme générant une analyse hebdomadaire
19/05	Révision du code pour optimiser ses performances
26/05	Mise en commun avec M.Morales Dussan
01/09	Présentation du travail final et rédaction du commentaire sur les programmes

8 Références

Vincent Le Goff : « Apprenez à programmer en Python »
https ://openclassrooms.com/fr/courses/235344-apprenez-a-programmer-en-python, Mis à jour le 29/07/2019

Chantal Gribaumont : « Administrez vos bases de données avec MySQL »
https ://openclassrooms.com/fr/courses/1959476-administrez-vos-bases-de-donnees-avec-mysql, Mis à jour le 03/09/2019

Ali Neishabouri : « Découvrez les librairies python pour la Data Science »
https ://openclassrooms.com/fr/courses/4452741-decouvrez-les-librairies-python-pour-la-data-science, Mis à jour le 05/08/2019

Site internet de l'Utadeo :
https ://www.utadeo.edu.co/es

Page Wikipédia de Jorge Tadeo Lozano :
https ://es.wikipedia.org/wiki/Jorge_Tadeo_Lozano

Maps of the world :
https ://www.mapsofworld.com/colombia/maps/colombia-political-map.jpg

9 Annexe